

Real-Time Error Protection of Embedded Codes for Packet Erasure and Fading Channels

Vladimir M. Stanković, Raouf Hamzaoui, and Zixiang Xiong, *Senior Member, IEEE*

Abstract—Reliable real-time transmission of packetized embedded multimedia data over noisy channels requires the design of fast error control algorithms. For packet erasure channels, efficient forward error correction is obtained by using systematic Reed–Solomon (RS) codes across packets. For fading channels, state-of-the-art performance is given by a product channel code where each column code is an RS code and each row code is a concatenation of an outer cyclic redundancy check code and an inner rate-compatible punctured convolutional code. For each of these two systems, we propose a low-memory linear-time iterative improvement algorithm to compute an error protection solution. Experimental results for the two-dimensional and three-dimensional set partitioning in hierarchical trees coders showed that our algorithms provide close to optimal average peak signal-to-noise ratio performance, and that their running time is significantly lower than that of all previously proposed solutions.

Index Terms—Embedded codes, image compression, joint source-channel coding, unequal error protection.

I. INTRODUCTION

THE design of robust systems for the transmission of images and video over unreliable communication channels has become a major research topic over the last years. Since error control techniques based on automatic repeat request (ARQ) introduce delays that may be unacceptable in real-time applications, many researchers proposed systems based on forward error correction (FEC) only [1]–[6]. The performance of these systems is optimized by finding an allocation of the transmission bit budget between the source and the channel coders that minimizes the expected distortion. However, for real-time applications, e.g., live video streaming, the bit allocation solution must be computed online to account for the time-varying nature of the source and channel statistics. Moreover, memory limitations also play an essential role in algorithm design. This paper proposes for two state-of-the-art FEC systems bit allocation algorithms satisfying these demands.

In the first part of the paper, we consider a transmission system for packet erasure channels. The system, which was introduced in [7], transforms an embedded source bitstream into N packets of L symbols each such that information

layers of decreasing importance are protected with increasingly weaker maximum distance separable erasure codes (e.g., Reed–Solomon (RS) codes). The system offers the desirable property that the reconstruction quality gracefully degrades as the packet loss rate increases [5]. Several researchers devised efficient unequal loss protection solutions for this system. Mohr, Riskin, and Ladner [5] proposed a local search algorithm. In [8], they presented a faster algorithm. Given $p = LN$ distortion-rate points of the source coder, the algorithm first computes the h vertices of their convex hull. Then, an unequal loss protection solution is found in $O(hN \log N)$ time. The authors show that the solution is optimal if the distortion-rate function is convex and fractional bit allocation assignments are accepted. Puri and Ramchandran [4] provided a Lagrange multiplier-based algorithm that also finds an optimal solution under the above assumptions. The algorithm starts by computing the h vertices of the convex hull of p points of the operational distortion-rate curve. Then, after an $O(h)$ step, a solution is found after several $O(N)$ Lagrange iterations. Stockhammer and Buchner [9] presented an $O(N^2L^2)$ dynamic programming algorithm that is optimal if the operational distortion-rate function is convex and the packet loss probability is a monotonically decreasing function of the number of lost packets. Dumitrescu, Wu, and Wang [10] independently found the same algorithm. However, they showed that its complexity can be reduced to $O(NL^2)$. Moreover, they gave an $O(N^2L^2)$ algorithm that is optimal in the general case. The first contribution of our paper is an $O(NL)$ local search algorithm that starts from a solution that maximizes the expected number of received source bits and iteratively improves this solution. The algorithm is inspired by our previous work [11], which considers a joint source-channel coding system for a binary symmetric channel (BSC).

The second part of the paper deals with two other channel models. The first one is a combination of a packet erasure channel and a binary symmetric channel. It can describe situations where packets of data from a wireline network are sent to receivers over a wireless connection. The second one is a flat-fading Rayleigh channel, which is a good model for wireless mobile communication. The most powerful transmission systems over such channels are due to Sherwood and Zeger [3] and Sachs, Anand, and Ramchandran [6]. Both systems use an embedded source code and a product channel code. The row code of the product code is a concatenation of an outer cyclic redundancy check (CRC) code and an inner rate-compatible punctured convolutional (RCPC) code, while its column code is a systematic RS code. Both systems use equal error protection along the rows and unequal error protection along the columns. But whereas the system of [3] puts the earliest symbols of the embedded bitstream in the first rows, the system of [6] puts these symbols

Manuscript received September 2, 2002; revised June 17, 2003. The results of this paper were presented in part at IEEE International Conference on Image Processing, Rochester, NY, September 2002, and at the IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland, August 2002. This paper was recommended by Associate Editor H. Gharavi.

V. Stanković and Z. Xiong are with the Department of Electrical Engineering, Texas A & M University, College Station, TX 77843 USA (e-mail: stankovi@ee.tamu.edu; zx@ee.tamu.edu).

R. Hamzaoui is with the Department of Computer and Information Science, University of Konstanz, Konstanz D-78464, Germany (e-mail: hamzaoui@inf.uni-konstanz.de).

Digital Object Identifier 10.1109/TCSVT.2004.831964

in the first columns. As a consequence, the first system has a better progressive ability. On the other hand, the system of [6] offers a better reconstruction quality (see [6] for a comparison). In [6], the error protection solution for the product code was optimized by determining the best RS protection for each RCPC code rate. This is time-consuming when many RCPC code rates are allowed. In contrast, we provide a heuristic algorithm that jointly optimizes the RS and the RCPC protections.

The paper is organized as follows. In Section II, we state the optimal unequal loss protection problem as a combinatorial optimization problem and present a local search algorithm that finds a local optimal solution to this problem. In Section III, we explain the product code setting and describe our fast error protection solution. In Section IV, we discuss source and channel statistics estimation. In Section V, we compare the expected peak signal-to-noise ratio (PSNR) performance and the running time of our algorithms to those of the previous ones for two popular embedded wavelet coders: the set partitioning in hierarchical trees (SPIHT) [13] coder for images and the three-dimensional (3-D) SPIHT [14] coder for video sequences. The PSNR performance of our algorithms was similar to that of the state-of-the-art, while their running time was lower.

II. PACKET LOSS PROTECTION

A. Problem Statement

In this section, we introduce our terminology and state the packet loss protection problem as a combinatorial optimization problem. Our notation closely follows that of [10].

Suppose that an embedded source bitstream is to be protected and sent through a packet erasure channel as N packets of L symbols (for example, bytes) each. A popular forward error correction system [4], [5], [7], [8], [10] builds L segments S_1, \dots, S_L , each of which consists of $m_i \in \{1, \dots, N\}$ source symbols and protects each segment S_i with an (N, m_i) systematic RS code (see Table I). For each $i \in \{1, \dots, L\}$, let $f_i = N - m_i$ denote the number of RS redundant symbols that protect segment S_i . If n packets of N are erased, then the RS codes ensure that all segments that contain at most $N - n$ source symbols can be recovered. Thus, by adding the constraint $f_1 \geq f_2 \geq \dots \geq f_L$, if at most f_i packets are erased, then the receiver can recover at least the first i segments. In the following, we denote by \mathcal{F} the set of L -tuples (f_1, \dots, f_L) such that $f_1 \geq f_2 \geq \dots \geq f_L$ and $f_i \in \{0, \dots, N - 1\}$ for $i = 1, \dots, L$.

Let ϕ denote the operational distortion-rate function of the source coder and let X be the random variable whose value is the number of packets erased. For a given L -segment RS protection $F = (f_1, \dots, f_L) \in \mathcal{F}$, the expected distortion is

$$E[d](F) = \sum_{i=0}^L P_i(F) \phi(V_i(F)) \quad (1)$$

where

$$P_i(F) = \begin{cases} \text{Prob}(X > f_1), & \text{for } i = 0 \\ \text{Prob}(f_{i+1} < X \leq f_i), & \text{for } i = 1, \dots, L-1 \\ \text{Prob}(X \leq f_L), & \text{for } i = L \end{cases}$$

$V_0(F) = 0$, and for $i = 1, \dots, L$, $V_i(F)$ is the number of source symbols in the first i segments, that is, $V_i(F) = \sum_{k=1}^i m_k = iN - \sum_{k=1}^i f_k$.

TABLE I
PROTECTION WITH RS CODES. THERE ARE $N = 5$ PACKETS (ROWS) OF $L = 5$ SYMBOLS EACH. CELLS LABELED BY NUMBERS CONTAIN SUCCESSIVE INFORMATION SYMBOLS OF AN EMBEDDED SOURCE. X DENOTES AN RS REDUNDANT SYMBOL

1	2	3	6	9
x	x	4	7	10
x	x	5	8	11
x	x	x	x	12
x	x	x	x	x

Let $p_N(n)$ denote the probability that exactly n packets of N are erased during transmission. Then for $i = 1, \dots, L-1$, we have

$$P_i(F) = \begin{cases} 0, & \text{if } f_i = f_{i+1} \\ \sum_{n=f_{i+1}+1}^{f_i} p_N(n), & \text{otherwise.} \end{cases}$$

A *distortion-optimal* L -segment RS protection minimizes (1) over \mathcal{F} .

In [10], the expected distortion (1) was given in the equivalent form

$$c_N(N) \phi(V_0(F)) + \sum_{i=1}^L c_N(f_i) (\phi(V_i(F)) - \phi(V_{i-1}(F)))$$

where $c_N(k) = \sum_{n=0}^k p_N(n)$, $k = 0, \dots, N$; thus, $c_N(f_i)$ is the probability that the receiver correctly recovers segment S_i .

B. Local Search Algorithm

In this section, we propose a fast heuristic algorithm to minimize (1). We use a local search technique similar to the one developed in [11]. The basic idea is to start from a *rate-optimal* solution, that is, an L -segment RS protection $F = (f_1, \dots, f_L) \in \mathcal{F}$ for which the expected number of correctly received source symbols

$$E[r](F) = \sum_{i=0}^L P_i(F) V_i(F) \quad (2)$$

is maximum, and to iteratively improve it. Our approach is motivated by the following conjecture.

Conjecture 1: Suppose that the distortion-rate function ϕ is nonincreasing (resp. nondecreasing) and convex (resp. concave). Let T^* be a distortion-optimal L -segment RS protection and let R^* be a rate-optimal L -segment RS protection. Then $V_L(T^*) \leq V_L(R^*)$.

This conjecture is corroborated by all our experimental results, and it can be proved for $L = 1$ [12]. It states that a distortion-optimal protection uses more protection symbols than a rate-optimal one. This observation will be the basis for our local search algorithm. After defining the neighborhood of a solution as a set of solutions that provide a stronger protection than the current candidate, we start at a rate-optimal solution and search for the best candidate in its neighborhood. If this candidate is better than the current solution, we adopt it and repeat the search from the new solution. Otherwise, we stop. The following proposition shows that the computation of a rate-optimal solution is straightforward.

Proposition 1: With the above notations, a rate-optimal L -segment RS protection is the equal loss protection (f_r, \dots, f_r) , where

$$f_r = \arg \max_{i=0, \dots, N-1} (N-i) \sum_{n=0}^i p_N(n).$$

Proof: The result is a simple consequence of the equality

$$\begin{aligned} E[r](f_1, \dots, f_L) &= \sum_{i=1}^L P_i(F) \sum_{k=1}^i m_k \\ &= \sum_{j=1}^L m_j \sum_{i=0}^{f_j} p_N(i). \end{aligned}$$

We now specify the neighborhood of a solution.

Definition 1: Let $F = (f_1, \dots, f_L) \in \mathcal{F}$. The neighborhood of F denoted by $\mathcal{N}(F)$ consists of the solutions of the form (f_1+1, f_2, \dots, f_L) , $(f_1+1, f_2+1, \dots, f_L)$, \dots , $(f_1+1, f_2+1, \dots, f_{L-1}+1, f_L+1)$ that are included in \mathcal{F} .

For example, suppose that $L = 4$ and $N = 8$. Then the neighbors of $F = (6, 6, 5, 4)$ are the solutions $(7, 6, 5, 4)$, $(7, 7, 5, 4)$, $(7, 7, 6, 4)$, and $(7, 7, 6, 5)$. Note that the solutions that do not belong to \mathcal{F} are not considered. For example, the neighborhood of $F = (7, 6, 5, 4)$ is empty.

The following pseudocode summarizes our local search algorithm.

Algorithm LS1: Given positive integers N and L , the operational distortion-rate function ϕ , and the probabilities $p_N(n)$, $n = 0, \dots, N$, a local search solution F is computed as follows.

- 1) Use Proposition 1 to compute a rate-optimal L -segment RS protection R^* . Set $F = R^*$.
- 2) If $\mathcal{N}(F) = \emptyset$, stop.
- 3) Compute $C = \arg \min_{S \in \mathcal{N}(F)} E[d](S)$.
- 4) If $E[d](C) < E[d](F)$, set $F = C$ and go to Step 2. Otherwise stop.

In the worst case, our algorithm starts from the rate-optimal solution $(0, \dots, 0)$ and stops at $(N-1, f_2, \dots, f_L)$. The determination of a rate-optimal solution can always be done in $O(N)$ steps. The refinement process needs $L(N-1)+1$ computations and $L(N-1)$ comparisons of cost function (1). Thus, the overall worst-case complexity of our algorithm is $O(NL)$.

When we compute cost function (1) for the neighbors of a solution, we exploit the fact that two successive neighbors differ in only one segment. Thus, only two probabilities P_i have to be recomputed. For example, let $F_1 = (7, 6, 5, 4)$ and $F_2 = (7, 7, 5, 4)$. Then $P_0(F_2) = P_0(F_1)$, $P_3(F_2) = P_3(F_1)$, and $P_4(F_2) = P_4(F_1)$.

Because our algorithm exploits the assumption of the convexity of the operational distortion-rate function of the source coder, it may provide unsatisfactory results when the convexity assumption is severely violated. To overcome this problem, we propose for such cases to compute our solution by applying Algorithm LS1 to the piecewise affine function obtained by joining the points of the operational distortion-rate curve at which the distortion changes. An alternative would be to use the lower convex hull of the operational distortion-rate curve.

As in [11], [12], we can also derive a useful lower bound for the expected distortion of a distortion-optimal protection.

TABLE II

PRODUCT CODE STRUCTURE. THERE ARE $N = 5$ PACKETS OF $L = 10$ SYMBOLS EACH. CELLS LABELED BY NUMBERS CONTAIN SUCCESSIVE INFORMATION SYMBOLS OF AN EMBEDDED SOURCE. x DENOTES AN RS REDUNDANT SYMBOL, + A CRC SYMBOL, AND o AN RCPC REDUNDANT SYMBOL. THE RCPC CODE NEED NOT BE SYSTEMATIC

1	2	3	6	9	+	+	o	o	o
x	x	4	7	10	+	+	o	o	o
x	x	5	8	11	+	+	o	o	o
x	x	x	x	12	+	+	o	o	o
x	x	x	x	x	+	+	o	o	o

Proposition 2: Suppose that ϕ is nonincreasing and convex. Let T^* be a distortion-optimal L -segment RS protection and let R^* be a rate-optimal L -segment RS protection. Then $E[d](T^*) \geq \phi(E[r](R^*))$.

Proof: This is a simple consequence of Jensen's inequality and the monotonicity of ϕ . ■

III. PRODUCT CODE

A. Problem Statement

In this section, the embedded source code is protected with a two-dimensional (2-D) product code using a concatenated CRC/RCPC code as the row code and systematic RS codes as column codes [6]. The resulting bitstream is sent as N packets of L symbols each.

The embedded source code is first protected with RS codes as in Section II. Then the CRC symbols are added to each row. Finally, each row is encoded with the same RCPC code. Let $\mathcal{R} = \{r_1, \dots, r_m\}$ be the set of available RCPC code rates. For $r_i \in \mathcal{R}$, we denote by $L(r_i)$ the sum of the number of source symbols and RS redundant symbols used in a packet protected with r_i . Thus, we have $L(r_i)$ source segments $S_1, \dots, S_{L(r_i)}$, where segment S_j , $1 \leq j \leq L(r_i)$, consists of $m_j \in \{1, \dots, N\}$ source symbols that are protected by $f_j = N - m_j$ RS symbols (see Table II).

Packets are sent over a channel where both packet erasures and bit errors may happen. Each received packet is decoded with the RCPC decoder. If the CRC detects an error, then the packet is considered to be lost (we suppose that all errors can be detected). Suppose now that n packets of N are lost (i.e., either erased during transmission or received but not correctly decoded by the RCPC decoder), then the RS codes ensure that all segments that contain at most $N-n$ source symbols can be recovered. By adding the constraint $f_1 \geq f_2 \geq \dots \geq f_{L(r_i)}$, we guarantee that the receiver can decode at least the first j segments whenever at most f_j packets are lost. For example, suppose that RCPC code rate r_i is used and suppose that the channel is modeled as the concatenation of a BSC and a memoryless packet erasure channel with packet loss probability q . Then the probability that a packet is considered to be lost is $q' = q + (1-q)p(r_i)$, where $p(r_i)$ is the probability that a packet of length L protected with rate r_i cannot be correctly decoded with the RCPC decoder. In this case, the probability that exactly n packets of N are lost is $p_N(n) = \binom{N}{n} q'^n (1-q')^{N-n}$.

By analogy with Section II, we denote by \mathcal{F}_i , $i = 1, \dots, m$, the set of $L(r_i)$ -tuples $(f_1, \dots, f_{L(r_i)})$ such that $f_1 \geq f_2 \geq$

$\dots \geq f_{L(r_i)}$ and $f_j \in \{0, \dots, N-1\}$ for $j = 1, \dots, L(r_i)$. A product code solution (r_i, F) is given by an RCPC code rate $r_i \in \mathcal{R}$ and an $L(r_i)$ -segment RS protection $F \in \mathcal{F}_i$.

A *distortion-optimal product code* solution (r_i^*, F^*) is given by an RCPC code rate $r_i^* \in \mathcal{R}$ and an $L(r_i^*)$ -segment RS protection $F^* \in \mathcal{F}_{i^*}$ that solve the minimization problem

$$\min_{r_i \in \mathcal{R}, F \in \mathcal{F}_i} \sum_{k=0}^{L(r_i)} P_k(F) \phi(V_k(F)) \quad (3)$$

where P_k , ϕ , and $V_k(F)$ are defined as in Section II. Solving (3) by brute-force is impractical because the number of possible product code solutions is $\sum_{i=1}^m \binom{L(r_i)+N-1}{L(r_i)}$.

In [6], the authors use the Lagrange-based optimization algorithm of [4] to determine a near-optimal $L(r_i)$ -segment RS protection for each $i = 1, \dots, m$. The RS protection that yields the smallest expected distortion is selected. Even though the Lagrange-based optimization algorithm is fast, the overall optimization can be too expensive for real-time systems when the number of candidate RCPC code rates is large.

B. Joint Product Code Optimization

In this section, we present a fast heuristic method for problem (3). In contrast to [6], we do not try to minimize (1) for each RCPC code rate. Our approach is similar to the one of the previous section. We start from a *rate-optimal product code* solution, that is, one that solves the maximization problem

$$\max_{r_i \in \mathcal{R}, F \in \mathcal{F}_i} \sum_{k=0}^{L(r_i)} P_k(F) V_k(F).$$

Then we try to improve this solution by progressively increasing the total number of protection symbols. This is done by alternately applying the local search algorithm of the previous section (Algorithm LS1) and decreasing the RCPC code rate. When we decrease the RCPC code rate, we replace the RS protection by its *child* defined as follows.

Definition 2: Let r_i and r_j be two code rates in \mathcal{R} with $r_j < r_i$. Let $F_i = (f_1, \dots, f_{L(r_j)}, \dots, f_{L(r_i)}) \in \mathcal{F}_i$ be an $L(r_i)$ -segment RS protection. Then the $L(r_j)$ -segment RS protection $F_j = (f_1, \dots, f_{L(r_j)}) \in \mathcal{F}_j$ is called the *child* of F_i in \mathcal{F}_j .

Table III illustrates the procedure for the product code of Table II. Note how the total number of redundant symbols increases.

We also exploit the fact that if S is our current RS protection, then one can exclude all RCPC code rates for which the lower bound of Proposition 2 is greater than $E[d](S)$. Indeed, a distortion-optimal RS protection corresponding to one such code rate cannot be better than S .

The following pseudocode summarizes our algorithm.

Algorithm LS2: Given positive integers N and L , the operational distortion-rate function ϕ , the set of RCPC code rates \mathcal{R} , and the probabilities $p_N(n)$, $n = 0, \dots, N$, a product code solution (r^*, S^*) is computed as follows.

- 1) Set $n = 0$ and $\mathcal{R}_n = \mathcal{R}$. For each code rate $r_i \in \mathcal{R}_n$, compute a rate-optimal $L(r_i)$ -segment RS protection F_i .
- 2) Determine $r_{k_n} = \arg \max_{r_i \in \mathcal{R}_n} E[r](F_i)$.
- 3) Apply Algorithm LS1 to F_{k_n} (i.e., start Algorithm LS1 from Step 2 with $F = F_{k_n}$). This gives a solution S_{k_n} . Set $S^* = S_{k_n}$ and $r^* = r_{k_n}$.

TABLE III
PRODUCT CODE OBTAINED FROM TABLE II BY DECREASING THE RCPC CODE RATE. THE NEW RS PROTECTION (4,4,2) IS CALLED THE CHILD OF THE OLD ONE (4,4,2,2,1)

1	2	3	+	+	o	o	o	o	o
x	x	4	+	+	o	o	o	o	o
x	x	5	+	+	o	o	o	o	o
x	x	x	+	+	o	o	o	o	o
x	x	x	+	+	o	o	o	o	o

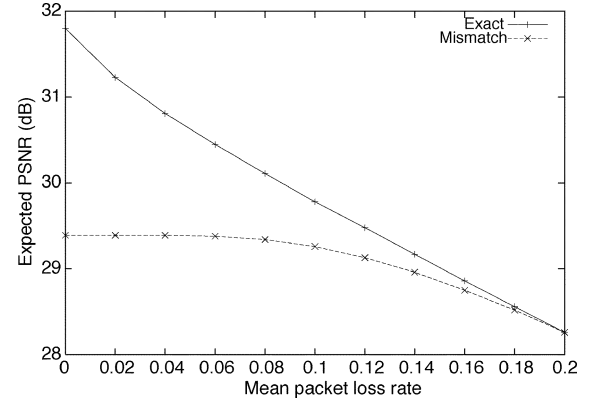


Fig. 1. Expected PSNR as a function of the mean packet loss rate for $N = 100$ packets of length $L = 48$ bytes for the RS system. The source data is the SPIHT bitstream of the 512×512 Lenna image. The upper curve shows the results when the error protection is computed at the exact channel conditions, while the lower curve shows the results when the error protection is computed for mean packet loss rate 0.2.

- 4) Find \mathcal{R}_{n+1} , the set of rates $r_i \in \mathcal{R}_n$ such that $i < k_n$ and $\phi(E[r](F_i)) \leq E[d](S^*)$. If $\mathcal{R}_{n+1} = \emptyset$, stop.
- 5) Let $r_{k_{n+1}}$ be the largest code rate in \mathcal{R}_{n+1} . Apply Algorithm LS1 to the child of S_{k_n} in $\mathcal{F}_{k_{n+1}}$. This gives a solution $S_{k_{n+1}}$. If $E[d](S_{k_{n+1}}) < E[d](S^*)$, set $S^* = S_{k_{n+1}}$ and $r^* = r_{k_{n+1}}$.
- 6) Set $n = n + 1$ and go to Step 4.

In the worst case, Algorithm LS2 computes for each $r_i \in \mathcal{R}$ the cost function (1) $(N-1)L(r_i) + 1$ times.

IV. SOURCE AND CHANNEL STATISTICS

The optimization of cost functions (1) and (2) assumes the availability of the channel statistics. A simple approach is to optimize for the worst possible channel condition. However, this often leads to overprotection and significantly reduces the performance. For example, Fig. 1 compares the performance of the local search algorithm of Section II-B when the solution is computed for the worst channel condition to its performance when the solution is computed for the exact channel condition. The experiment shows that a mismatch between the optimization settings and the actual statistics can lead to a significant drop in quality. However, when a feedback channel is available, which is the typical situation in practice, the channel can be monitored (see [15] and [16] for channel estimation), allowing the transmitter to optimize the cost functions online with up-to-date channel statistics.

TABLE IV

RS SYSTEM. CPU TIME IN SECONDS AND EXPECTED PSNR IN DECIBELS FOR N PACKETS OF $L = 48$ BYTES EACH. Da DENOTES THE OPTIMAL ALGORITHM OF [10], Db DENOTES THE SECOND ALGORITHM OF [10], Ma DENOTES THE ALGORITHM OF [5], Mb DENOTES THE ALGORITHM OF [8], P DENOTES THE ALGORITHM OF [4], AND LS1 IS THE LOCAL SEARCH ALGORITHM OF SECTION II. THE SOURCE CODE IS THE SPIHT BITSTREAM OF THE 512×512 LENA IMAGE. THE CHANNEL IS A PACKET ERASURE CHANNEL WITH MEAN PACKET LOSS RATE 0.2

N	Da [10]		Db [10]		Ma [5]		Mb [8]		P [4]		LS1	
	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
100	28.27	1.51	28.27	0.25	28.26	0.33	28.26	0.05	28.24	0.03	28.26	< 0.01
200	30.99	8.85	30.99	0.54	30.95	1.49	30.97	0.10	30.95	0.06	30.93	< 0.01
400	33.90	35.96	33.90	1.24	33.84	4.29	33.87	0.33	33.88	0.15	33.84	< 0.01
600			35.62	1.85	35.37	8.12	35.60	0.72	35.57	0.24	35.58	0.01
800			36.84	2.70	36.65	13.36	36.81	1.33	36.79	0.32	36.79	0.01
1000			37.86	3.27	37.18	14.20	37.85	2.15	37.81	0.41	37.84	0.02

The optimization of the cost functions also requires the source statistics, which may not be available offline. This is the case in many applications, including live video streaming where the source is constantly changing and the situation where a mobile phone equipped with a digital camera takes a picture and immediately sends it to a receiver. A problem is that computing the exact operational distortion-rate curve of the source coder is not possible under the real-time constraint. For embedded wavelet coders such as SPIHT, JPEG2000 [17], and 3-D SPIHT a solution may consist of estimating the distortion-rate points in the wavelet domain during the encoding (see [13] for SPIHT). An alternative is to quickly compute a few distortion-rate points, fit them to a parametric model, and use the model instead of the true distortion-rate function [18].

V. RESULTS

A. Packet Loss Protection

In this section, we first present optimization results for the system of Section II. We compare the time complexity and the expected PSNR performance of our local search algorithm to that of the following algorithms.

- 1) The local search algorithm of Mohr *et al.* [5]. The quality of the solution computed by this algorithm is dependent on a search parameter Q that has to be fixed ahead of time. However, there is a trade-off between quality and computing time. Since time is a major issue in our comparison, we used the algorithm with $Q = 1$, which generally yields the fastest results.
- 2) The convex hull-based algorithm of Mohr *et al.* [8].
- 3) The Lagrangian-based algorithm of Puri and Ramchandran [4].
- 4) The two algorithms of Dumitrescu *et al.* [10]. Note that one of these algorithms finds an optimal solution.

The C-codes of the algorithms of [4], [8], and [10] were provided by the authors of these papers. We implemented the algorithm of [5]. All programs were run on a PC having an AMD Athlon (TM) XP 1600 1400 MHz processor with a main memory size of 1 Gbyte. In all experiments, N packets of L bytes each were sent over a packet erasure channel. Unless otherwise mentioned, the packet loss probability function $p_N(n)$ was modeled as an exponentially decreasing function of n with a given mean loss rate M , that is, $p_N(n) = Ae^{-an/N}$ where A and a were chosen such

that $\sum_{n=0}^N p_N(n) = 1$ and $\sum_{n=0}^N np_N(n) = MN$. We focus on packet payload size $L = 48$ bytes, which is the standard in ATM networks, but we also give results for larger packet sizes. The goal of the algorithms was to maximize the expected PSNR. Thus, instead of cost function (1), we used the objective function $\sum_{i=0}^L P_i(F) \text{PSNR}(V_i(F))$, where $\text{PSNR}(V_i(F))$ is the PSNR corresponding to $V_i(F)$ symbols of the source code.

Both [8] and [4] compute the vertices of the convex hull of the distortion-rate points in a preprocessing step. Whereas [4] uses Jarvis's march [19], which runs in $O(ph)$ time, where $p = LN$ is the number of distortion-rate points and h is the number of vertices of the convex hull, [8] uses the incremental method [19] whose $O(p \log p)$ complexity reduces to $O(p)$ because the input rates are already sorted. But since these are asymptotic results, one cannot affirm that the incremental method is always faster.

Table IV shows the expected PSNR in dB and the CPU time in seconds of the algorithms for the SPIHT bitstream of the 8 bits per pixel (bpp) gray-scale 512×512 Lenna image. Table V shows the results for the 3-D SPIHT bitstream of the Y component of the 256-frame 176×144 Foreman video sequence. In both tables, $L = 48$ bytes, and the mean loss rate M is 0.2. The CPU times take into account all operations required by an algorithm. This includes computing the convex hull for NL points of the PSNR-rate curve in algorithms [4], [8]. However, for all algorithms, we did not include the preprocessing time used to store the file containing the operational PSNR-rate points in an array.

The tables show that our local search algorithm was faster than all previous algorithms. For the SPIHT bitstream, for example, it was up to 32 times faster than the previous fastest solution, the algorithm of [4]. On the other hand, the PSNR performance of our solution was similar to that of the best previous algorithms, including the optimal algorithm of [10]. However, since this algorithm has huge memory requirements: $N^2 L^2$ floating point numbers and $N^2 L^2$ integers, our computer ran out of memory when N or L were large, and we were not able to report the results. The local search algorithm of [5] was much slower than our algorithm, and it yielded a poor solution for many large values of N . When this algorithm was used with the search parameter Q set to N (this setting gives the highest PSNR), it produced a slightly better expected PSNR than our algorithm, but the gain did not exceed 0.08 dB, and its computing time was up to 1000 times higher than that of our algorithm. The optimal algorithm of [10] was the slowest.

TABLE V
RS SYSTEM. CPU TIME IN SECONDS AND EXPECTED PSNR IN DECIBELS FOR N PACKETS OF $L = 48$ BYTES EACH. THE SOURCE CODE IS THE 3-D SPIHT BITSTREAM OF THE FOREMAN SEQUENCE. THE CHANNEL IS A PACKET ERASURE CHANNEL WITH MEAN PACKET LOSS RATE 0.2

N	Db [10]		Ma [5]		Mb [8]		P [4]		LS1	
	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
400	21.90	1.29	21.82	2.35	21.92	0.12	21.86	0.04	21.92	< 0.01
600	23.08	1.99	22.88	7.62	23.06	0.19	23.05	0.07	23.04	0.01
800	23.81	2.79	23.31	7.11	23.80	0.28	23.78	0.10	23.77	0.01
1000	24.41	3.61	24.37	22.28	24.39	0.37	24.38	0.13	24.34	0.01
1200	24.98	4.68	24.75	37.61	24.94	0.57	24.92	0.17	24.92	0.01
1400	25.41	5.38	24.84	31.82	25.40	0.78	25.36	0.24	25.36	0.02
1600	25.80	6.51	25.04	28.33	25.80	1.04	25.76	0.27	25.73	0.02

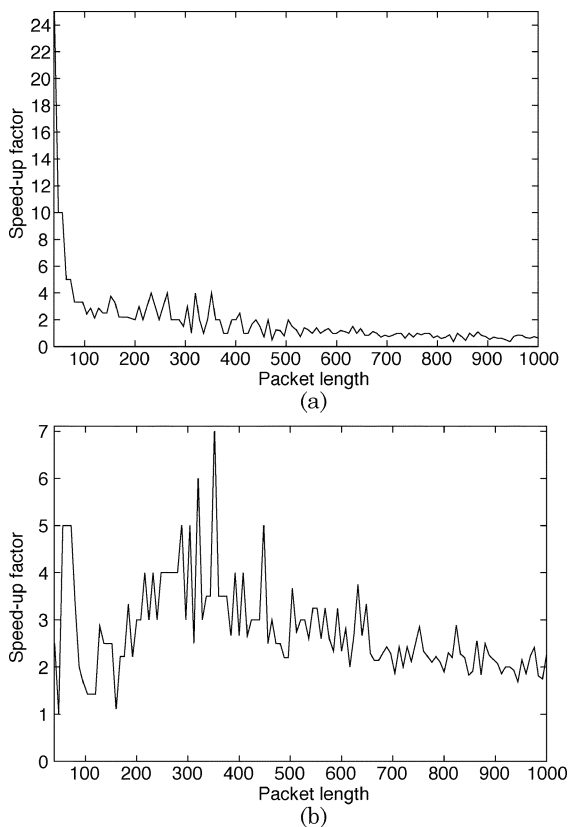


Fig. 2. Speed-up factor of our algorithm over the algorithm of [4] as a function of the packet length in bytes for the RS system. The source code is the 3-D SPIHT bitstream of the Foreman sequence. The number of packets is $N = 200$. The channel is a packet erasure channel with mean packet loss rate 0.05. The algorithm of [4] computes the convex hull of the PSNR-rate points with (a) the incremental method and (b) Jarvis's march.

In the next experiment, we study the influence of the packet size L on the efficiency of our algorithm. Fig. 2 shows the speed-up factor of our algorithm over the algorithm of [4] as a function of the packet size. In Fig. 2(a), the algorithm of [4] is used with Jarvis's march, while in Fig. 2(b) it is used with the incremental method. The experiment shows that the algorithm of [4] becomes faster than our algorithm (i.e., the speed-up factor is less than one) only when the packet size is large, and the fastest technique for computing the convex hull is selected.

Finally, Table VI shows results for a two-state Markov packet loss model, which adequately describes packet loss on the In-

ternet [20]. Details on how to compute the probability function $p_N(n)$ for this model can be found in [20]. In the simulations, we used an average loss probability of 0.1 and an average burst length of 9.57 [20]. Here again our algorithm was the fastest.

B. Product Code

We now present results for the optimization of the product code of Section III. We used a 16-bit CRC code with generator polynomial 0×15935 . The generator polynomials of the RCPC were (0117, 0127, 0155, 0171), the mother code rate was $1/4$, and the puncturing rate was 8. Thus, the set of RCPC code rates was $\{8/32, \dots, 8/9\}$. The decoding of the RCPC code was done with a list Viterbi algorithm where the maximum number of candidate paths was 100.

We compare the performance of two optimization approaches. The first one is Algorithm LS2 of Section III. The second approach [6] searches for an optimal RS protection for each RCPC code rate and selects the one with the highest expected PSNR. For a given RCPC code rate, the RS protection was determined with the Lagrange-based algorithm of [4], the algorithm of [8], and the two algorithms of [10] (Da and Db). Note that the product code is optimal when Da is used.

We first provide results for a channel modeled as a concatenation of a BSC and a packet erasure channel. The bit error rate (BER) of the BSC was 0.1, and the mean packet loss rate in the erasure channel was $M = 0.05$. All programs were run on the same machine as above. Here also the CPU times include all operations required by an algorithm.

Table VII shows the expected PSNR in decibels and the time in seconds versus the number of packets N for the SPIHT bitstream of the 8 bpp 512×512 Lenna. The length of a packet was equal to 48 bytes, including one byte for the header. We used only the 10 RCPC code rates $\{8/27, \dots, 8/18\}$ because the other 14 can be excluded a priori. Indeed, since the probability of decoding error of many code rates was zero, only the largest one among them was kept. We also ignored other code rates because their residual bit error rate was greater than 0.1.

Table VIII shows results for the 3-D SPIHT bitstream of the Y component of the 176×144 Foreman video sequence. Here the packet length was 1000 bytes. In this case, only the nine RCPC code rates $\{8/32, \dots, 8/24\}$ had to be used. The last column shows the results for the solution obtained by applying our algorithm to the piecewise affine function associated to the PSNR-rate curve (see

TABLE VI
RS SYSTEM. CPU TIME IN SECONDS AND EXPECTED PSNR IN DECIBELS FOR N PACKETS OF $L = 48$ BYTES EACH. THE SOURCE CODE IS THE SPIHT BITSTREAM OF THE 512×512 LENA IMAGE. THE CHANNEL IS A TWO-STATE MARKOV PACKET LOSS MODEL WITH AVERAGE LOSS PROBABILITY 0.1 AND AVERAGE BURST LENGTH 9.57

N	Da [10]		Db [10]		Ma [5]		Mb [8]		P [4]		LS1	
	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
100	29.68	1.73	29.68	0.25	29.65	0.17	29.67	0.05	29.66	0.02	29.67	< 0.01
200	33.03	10.54	33.03	0.54	33.00	0.34	33.03	0.08	33.03	0.06	33.02	< 0.01
400			36.48	1.18	36.46	1.03	36.47	0.18	36.47	0.14	36.47	< 0.01
600			38.47	1.88	38.47	2.06	38.46	0.30	38.46	0.23	38.46	< 0.01
800			39.92	2.47	39.87	2.02	39.92	0.47	39.91	0.31	39.91	< 0.01
1000			41.23	3.26	41.22	2.68	41.23	0.60	41.23	0.40	41.22	< 0.01

TABLE VII
PRODUCT CODE SYSTEM. CPU TIME IN SECONDS AND EXPECTED PSNR IN DECIBELS FOR N PACKETS OF $L = 48$ BYTES EACH. THE ABBREVIATIONS Da, Db, P, AND M DENOTE THE APPROACH OF [6] WHEN THE RS PROTECTION FOR EACH RCPC CODE RATE IS DETERMINED WITH THE OPTIMAL ALGORITHM OF [10], ALGORITHM B OF [10], THE ALGORITHM OF [4], AND THE ALGORITHM OF [8], RESPECTIVELY. LS2 IS THE ALGORITHM OF SECTION III. THE SOURCE CODE IS THE SPIHT BITSTREAM OF THE 512×512 LENA IMAGE. THE CHANNEL IS A CONCATENATION OF A PACKET ERASURE CHANNEL WITH MEAN LOSS RATE 0.05 AND A BSC WITH BER 0.1

N	Da [10]	Db [10]		P [4]		M [8]		LS2	
	PSNR	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
100	26.16	26.14	0.20	26.14	0.04	26.15	0.06	26.14	0.01
200	28.53	28.53	0.45	28.50	0.07	28.51	0.19	28.51	0.01
400	31.30	31.30	0.96	31.26	0.15	31.29	0.76	31.27	0.03
600	33.07	33.07	1.52	33.03	0.21	33.04	1.96	33.03	0.05
800	34.33	34.33	2.74	34.28	0.29	34.31	3.85	34.26	0.06
1000		35.34	3.57	35.28	0.40	35.32	6.66	35.31	0.08

TABLE VIII
PRODUCT CODE SYSTEM. CPU TIME IN SECONDS AND EXPECTED PSNR IN DECIBELS FOR N PACKETS OF $L = 1000$ BYTES EACH. LS2+PA MEANS THAT IN ALGORITHM LS2, ALGORITHM LS1 WAS APPLIED TO THE PIECEWISE AFFINE FUNCTION ASSOCIATED TO THE PSNR-RATE CURVE. THE SOURCE CODE IS THE 3-D SPIHT BITSTREAM OF THE FOREMAN SEQUENCE. THE CHANNEL IS A CONCATENATION OF AN ERASURE CHANNEL WITH MEAN PACKET LOSS RATE 0.05 AND A BSC WITH BER 0.1

N	P [4]		M [8]		LS2		LS2+PA	
	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
100	24.79	0.15	24.80	0.22	24.75	0.04	24.75	0.04
200	26.80	0.34	26.80	0.48	26.79	0.08	26.79	0.09
400	29.04	0.66	29.04	1.06	28.95	0.14	28.99	0.19
600	30.47	0.89	30.47	2.03	30.38	0.18	30.42	0.30
800	31.61	1.20	31.61	3.41	31.52	0.26	31.56	0.52
1000	32.58	1.57	32.58	4.98	32.45	0.31	32.51	0.63

the end of Section II). The expected PSNR, however, is given for the true PSNR-rate curve. In the algorithms of [4] and [8], we used the incremental method for the computation of the convex hull since this method yields the fastest results when L is large.

Table IX shows results for a Rayleigh channel where the average signal-to-noise ratio (SNR) was 10 dB, and the normalized Doppler spread f_D was 10^{-5} . The probability function $p_N(n)$ was simulated with Jakes's method [21]. Here all 24 RCPC code rates were used.

In all cases, the expected PSNR performance of our solution was almost optimal. Moreover, our algorithm was significantly faster than the previous approach. For example, for the Rayleigh

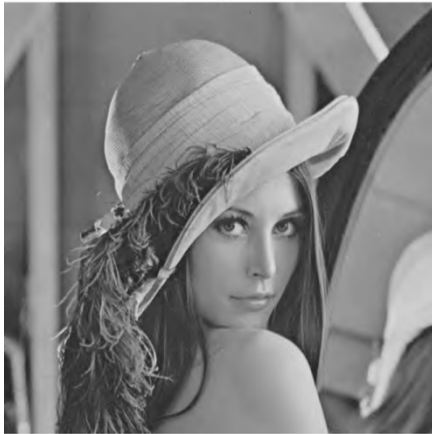
channel and 3-D SPIHT, our algorithm was 20 times faster than the approach of [6] with the algorithm of [8] and about 10 times faster when the algorithm of [4] was used. Note that since L is large, the second algorithm of [10], which needs the storage of N^2L floating point numbers and N^2L integers, could not be used for $N = 200$. The bound of Proposition 2, which is exploited in Step 4 of our algorithm, was very helpful. For the Rayleigh channel, it allowed us to exclude 19 code rates for SPIHT and 22 code rates for 3-D SPIHT. In all experiments, the RCPC code rate selected by our algorithm was either the one corresponding to the rate-optimal product code solution (Step 2), or the next tested one.

TABLE IX
PRODUCT CODE SYSTEM. CPU TIME IN SECONDS AND EXPECTED PSNR IN DECIBELS FOR 200 PACKETS OF L BYTES EACH, WHERE $L = 48$ FOR SPIHT AND 1000 BYTES FOR 3-D SPIHT. THE CHANNEL IS A RAYLEIGH FADING CHANNEL WITH $\text{SNR} = 10$ dB AND $f_D = 10^{-5}$

Source coder	N	Da [10]	Db [10]		P [4]		M [8]		LS2	
		PSNR	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
SPIHT	200	31.01	31.01	2.14	30.98	0.15	31.00	0.45	30.95	0.02
3D SPIHT	200				29.93	0.77	29.93	1.44	29.93	0.07



(a)



(b)

Fig. 3. RS system. Expected reconstruction of the 512×512 Lenna image for a packet erasure channel with a mean packet loss rate of 0.2. Results are given at transmission rate (a) 0.146 bpp and (b) 0.879 bpp. The source coder is SPIHT.

Figs. 3 and 4 show the reconstruction quality of the Lenna image at various transmission rates for the expected PSNR given by our solutions.

We repeated all above experiments for two other standard images (Goldhill and Barbara), many other channel conditions, another source coder (JPEG2000) and obtained similar results.

VI. CONCLUSION

We proposed two heuristic algorithms for unequal error protection of embedded multimedia data in unreliable channels. Both algorithms have very low memory requirements and a linear worst-case time complexity, making them suitable for real-time applications.



(a)



(b)

Fig. 4. Product code system. Expected reconstruction of the 512×512 Lenna image at transmission rate 0.292 bpp for (a) concatenation of a BSC with BER 0.1 and a packet erasure channel with a mean packet loss rate of 0.05 and (b) Rayleigh fading channel with $\text{SNR} = 10$ dB and $f_D = 10^{-5}$. The source coder is SPIHT.

The first algorithm provides RS-based unequal loss protection in packet erasure channels. It starts with a linear-time complexity step that maximizes the average number of correctly decoded source bits, followed by a quick local refinement of the resulting equal error protection solution to minimize the average distortion. For the SPIHT, JPEG2000, and 3-D SPIHT source coders, our algorithm achieved state-of-the-art average PSNR performance. Moreover, its time complexity was lower than that of the best previous algorithms, when the packet length was not too large.

The second algorithm is based on a fast joint optimization technique that provides a product code protection for transmis-

sion over wireless channels. Our algorithm was faster than the original approach of [6], even when the packet size was large.

ACKNOWLEDGMENT

The authors would like to thank S. Dumitrescu, X. Wu, and Z. Wang for providing them with the C-codes of their algorithms [10] and D. Sachs and K. Ramchandran for sending them the C-code used in [4]. They also thank A. Mohr, R. Ladner, and E. Riskin for making the C-code of algorithm [8] available. Finally, they thank G. Constantinescu for her work in programming.

REFERENCES

- [1] G. Davis and J. Danskin, "Joint source and channel coding for image transmission over lossy packet networks," in *Proc. SPIE Conf. Wavelet Applications of Digital Image Processing XIX*, Denver, CO, Aug. 1996, pp. 376–387.
- [2] P. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Process. Lett.*, vol. 4, pp. 191–198, July 1997.
- [3] —, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Trans. Commun.*, vol. 46, pp. 1555–1559, Dec. 1998.
- [4] R. Puri and K. Ramchandran, "Multiple description coding using forward error correction codes," in *Proc. 33rd Asilomar Conf. Signals and Systems*, vol. 1, Pacific Grove, CA, Oct. 1999, pp. 342–346.
- [5] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Select. Areas Commun.*, vol. 18, no. 7, pp. 819–828, Dec. 2000.
- [6] D. G. Sachs, R. Anand, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated codes," in *Proc. SPIE*, vol. 3974, Jan. 2000, pp. 300–311.
- [7] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, no. 42, pp. 1737–1744, Nov. 1996.
- [8] A. Mohr, R. Ladner, and E. Riskin, "Approximately optimal assignment for unequal loss protection," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Vancouver, BC, Canada, Sept. 2000, pp. 367–370.
- [9] T. Stockhammer and C. Buchner, "Progressive texture video streaming for lossy packet networks," presented at the *PV 11th Int. Packet Video Workshop*, Kyongju, Korea, May 2001.
- [10] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," in *Proc. Data Compression Conf.*, Snowbird, UT, Apr. 2002, pp. 73–82.
- [11] R. Hamzaoui, V. Stanković, and Z. Xiong, "Rate-based versus distortion-based optimal joint source-channel coding," in *Proc. Data Compression Conf.*, Snowbird, UT, Apr. 2002, pp. 63–72.
- [12] R. Hamzaoui, V. Stanković, and Z. Xiong, (2003, Oct.) Rate-based versus distortion-based optimal joint source-channel coding. *Konstanzer Schriften in Mathematik Informatik* [Online] Preprint no. 194, <http://www.inf.uni-konstanz.de/Preprints/preprints-all.htm>
- [13] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.
- [14] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 1365–1374, Dec. 2000.
- [15] S. Savage, "Sting: a TCP-based network measurement tool," in *Proc. USENIX Symp. Internet Technologies and Systems*, Oct. 1999, pp. 71–79.
- [16] H. Sanneck and G. Carle, "A framework model for packet loss metrics based on loss runlengths," in *Proc. SPIE/ACM SIGMM Multimedia Computing and Networking Conf.*, San Jose, CA, Jan. 2000.
- [17] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Norwell, MA: Kluwer, 2002.
- [18] Y. Charfi, R. Hamzaoui, and D. Saupe, "Model-based real-time progressive transmission of images over noisy channels," in *Proc. IEEE Wireless Communications and Networking Conf.*, vol. 2, New Orleans, LA, Mar. 2003, pp. 784–789.
- [19] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry*, 2nd ed. Berlin, Germany: Springer Verlag, 2000.
- [20] U. Horn, K. Stuhlmüller, M. Link, and B. Girod, "Robust internet video transmission based on scalable coding and unequal error protection," *Image Communication*, vol. 15, no. 1–2, pp. 77–94, Sept. 1999.
- [21] W. C. Jakes, *Microwave Mobile Communications*. New York: Wiley, 1974.



Vladimir M. Stanković received the Dipl.-Ing. degree in electrical engineering from the University of Belgrade, Serbia, in 2000, and the Dr.-Ing. degree from the University of Leipzig, Leipzig, Germany, in 2003.

From 2002 to 2003, he was with the Department of Computer and Information Science, University of Konstanz, Konstanz, Germany. He is currently a Post-doctoral Research Associate with the Department of Electrical Engineering of Texas A & M University, College Station. His research interests include robust

image and video transmission, error protection coding, and data compression.



Raouf Hamzaoui received the Maîtrise de mathématiques from the Faculty of Sciences of Tunis, Tunisia, in 1986, the M.Sc. degree in mathematics from the University of Montreal, Montreal, QC, Canada, in 1993, and the Dr. rer. nat. degree from the Faculty of Applied Sciences of the University of Freiburg, Freiburg, Germany, in 1997.

From 1998 to 2002, he was a Research Assistant with the Computer Science Department of the University of Leipzig, Leipzig, Germany. He is currently a Research Assistant with the Department of Computer and Information Science of the University of Konstanz, Konstanz, Germany. His research interests include data compression and joint source-channel coding.



Zixiang Xiong (S'91–M'96–SM'02) received the Ph.D. degree in electrical engineering in 1996 from the University of Illinois at Urbana-Champaign.

From 1997 to 1999, he was with the University of Hawaii, Honolulu. Since 1999, he has been with the Department of Electrical Engineering, Texas A & M University, where he is an Associate Professor. He spent the summers of 1998 and 1999 at Microsoft Research, Redmond, WA, and the summers of 2000 and 2001 at Microsoft Research in Beijing, China.

His current research interests are joint source-channel coding, distributed source coding, genomic signal processing and network traffic smoothing and modeling.

Dr. Xiong received an National Science Foundation Career Award in 1999, an Army Research Office (ARO) Young Investigator Award in 2000 and an Office of Naval Research (ONR) Young Investigator Award in 2001. He also received an Eugene Webb Faculty Fellow Award in 2001 and a Select Young Faculty Award in 2002, both from Texas A&M University. He is currently an associate editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and IEEE TRANSACTIONS ON IMAGE PROCESSING.